

## Лабораторная работа 9.

Черновик 0.55

Целью работы является знакомство студентов со строками и их обработкой.

Студенты должны получить и закрепить на практике следующие знания и умения:

1. Работа с динамической памятью.
2. Обработка строк.
3. Обработка текстовых файлов.
4. Организация корректной работы с ресурсами (динамически выделенная память, файловые дескрипторы).
5. Использование в программе аргументов командной строки.
6. Контроль правильности работы с динамической памятью с помощью специального ПО.

### 1. Общее задание

1. Исходный код лабораторной работы располагается в ветке lab\_09. В этой ветке создается папка lab\_09\_1, в которой располагается решение первой задачи, и папка lab\_09\_2 – для исходного кода второй задачи.
2. При решении обеих задач запрещается использовать любые стандартные функции для обработки строк.
3. Исходный код должен соответствовать правилам оформления исходного кода.
4. Решение каждой задачи оформляется как многофайловый проект.
5. Для первой задачи реализуются только модульные тесты. Для второй задачи реализуются как модульные, так и функциональные тесты. Входные данные должны располагаться в файлах in\_z.txt (возможно таких файлов будет несколько), выходные out\_z.txt, где z – номер тестового случая. Тестовые данные готовятся и помещаются под версионный контроль еще до того, как появится реализация задачи.

### 2. Индивидуальное задание

#### Задача 1

Реализовать указанные строковые функции.

1. strpbrk
2. strspn
3. strcspn
4. strchr
5. strrchr
6. strdup
7. strndup

Варианты заданий распределяются преподавателем. Имена функций, которые реализуются самостоятельно, начинаются с префикса “my\_” (если нужно реализовать функцию strpbrk, в программе она должна называться my\_strpbrk).

Эта задача подразумевает только реализацию указанных функций и модульных тестов к ним (т.е. никакого отдельного приложения для работы с реализованными функциями можно не

писать). При этом в модульных тестах нужно сравнить поведение (спецификацию) своей функции и соответствующей стандартной.

#### *Замечание 1*

Функция `strdup` становится доступной только при указании стандарта `gnu99` (т.е. `-std=gnu99`).

#### *Замечание 2*

Функция `strndup` в библиотеке компилятора MinGW не реализована. Для ее тестирования пишутся модульные тесты, сравнение с реализацией из библиотеки не выполняется.

#### *Замечание 3.*

Для компилятора `gcc` из MSYS2 «замечание 2» не проверялось.

### *Задача 2*

Необходимо реализовать следующие функции:

1. Чтения строки из текстового файла. Строка должна читаться целиком (т.е. до символа EOL или EOF) за один вызов функции.

```
ssize_t getline(char **lineptr, size_t *n, FILE *stream);
```

Справка по функциям (англ.)

<http://pubs.opengroup.org/onlinepubs/9699919799/functions/getline.html>

Для реализации этой функции запрещается использовать посимвольное чтение из файла. Вместо этого, следует завести статический или динамический буфер небольшого (например, 128 байт) размера, чтение в этот буфер выполнять с помощью `fgets`, а прочитанные данные из буфера переписывать в результирующую строку. Память под результирующую строку можно выделять с помощью функции `realloc`.

Функция должна называться “`my_getline`”.

2. Замены в строке указанной непустой подстроки произвольной длины на другую подстроку произвольной длины (возможно пустую). Память под результирующую строку выделяется динамически.

```
char* str_replace(const char *source, const char *search, const char *replace);
```

С помощью реализованных функций обработать текстовый файл, в котором выполнить замену всех вхождений указанной подстроки на заданную. В файле необходимо обработать все строки. Преобразованные строки записать в новый файл.

#### *Пример*

В строке “aaaa” заменяем подстроку “aa” на “bbb”. После замены получаем “bbbbbb”.

Все параметры передаются как аргументы командной строки.

```
app.exe in.txt out.txt -s search -r replace
```

Замечание 4.

Что такое POSIX и для чего он нужен

- <https://ru.wikipedia.org/wiki/POSIX>
- <https://en.wikipedia.org/wiki/POSIX>
- <https://stackoverflow.com/questions/1780599/i-never-really-understood-what-is-posix>

То, что интересно нам в POSIX

- <http://pubs.opengroup.org/onlinepubs/9699919799>

Для удобства использования можно доставить «справку» по POSIX функциям в MSYS2:

```
pacman -S man-pages-posix
```

после установки

```
man strspn  
man qsort  
  
и т.д.
```